International Islamic University Chittagong

Department of Computer Science & Engineering

Final Examination, Autumn 21

Course Code:CSE-3527 Course Title: Compiler

Total marks: ***50*** Time: ***2.5*** hours

[Answer all the questions; in some questions, there are options; solve the one you have been instructed to solve.]

Group-A

1.a)	Show the model of a nonrecursive predictive parser.	**3**	**CO1**
1.b)	Define Left factoring and left recursion. Consider the following Grammar: Q->QED |q E->e N->NFA |n D->DFA |d F->f A->Ad| Ac| ad| ac After removing the immediate left recursion what will be the grammar?	**5**	**CO1**
	Or		
1.b)	Consider the following Grammar: S->aAR|Ra A->S R>r|€ Compute first and follow for each of the grammar. Is the grammar a LL (1) grammar?	**5**	**CO1**
1.c)	Write down the steps of top down parsing.	**2**	**CO1**
2.a)	What is the relation between left recursion and left factoring? Eliminate left recursion from the following grammar if needed. S → A A → Ad / Ae / aB / ac B → bBc / f	**5**	**CO3**
	Or		
2.a)	Make the parsing table for this following grammar S->iEtS|iEtSeS|a E->b	**5**	**CO3**
2.b)	Compute FIRST and FOLLOW for the following grammar. Also Parse the following string using the LL(1) parsing table for the following grammar. S → A A → aB / Ad B → b C → g	**5**	**CO3**

Group-B

3.a)	Write some examples of top-down parsing and bottom-up parsing. Consider the following grammar- E → E – E E → E x E E → id Parse the input string id – id x id using a shift-reduce parser.	**5**	**CO3**

3.b)	Define the synthesized attributes and inherited attributes with example. Using syntax-directed definition draw the annotated parse tree and dependency graph for the sentence **(3*5+2)*2n.** S →En E →E_1 + T E →T T → T_1 * F T → F F →(E) F →digit	**5**	**CO3**
4.a)	Describe Peephole Optimization	**2.5**	**CO3**
	Translate the arithmetic expression a*- (b+c).	**5**	**CO3**
4.b)	i)Syntax Tree ii)Quadruples		
	Or		
4.b)	Translate the arithmetic expression a*a- (b+c)/d i) Triples ii) indirect triples	**5**	**CO3**
4.c)	Draw the DAG for the expression a+a*(b-c)+(b-c)*d	**2.5**	**CO3**
5.a)	Find out the basic block and draw the flow graph for this three address code.	**4**	**C(**

1)i=m-1
2)j=n
3)t1=4*n
4)v=a[t1]
5)i=i+1
6)t2=4*i
7)t3=a[t2]
8)if t3<v goto (5)
9)j=j-1
10)t4=4*j
11)t5=a[t4]
12)if t5>v goto(9)
13)if i>=j goto (23)
14)t6=4* i
15)x=a[t6]
16) t7=4*i
17)t8=4*j
18) t9=a[t8]
19) a[t7]=t9
20) t10=4*j
21) a[t10]=x
22) goto 5
23) t11=4*i
24) x=a[t11]
25) t12=4*i
26) t13=4*n
27) t14=a[t13]
28) a[12]=t14
29) t15=4*n
30) a[t15]=x

5.b)	Eliminate the local and global common sub expressionfrom the basic block of question 5(a) and draw the flow graph after elimination.	**5**	**CO**
	Or		
5.b)	Show the position of code generation in the compiler process. Briefly describe the issues of code generation.	**5**	**CO3**
5.c)	Write the algorithm to partitioning three address instructions into the basic blocks.	**1**	**CO3**